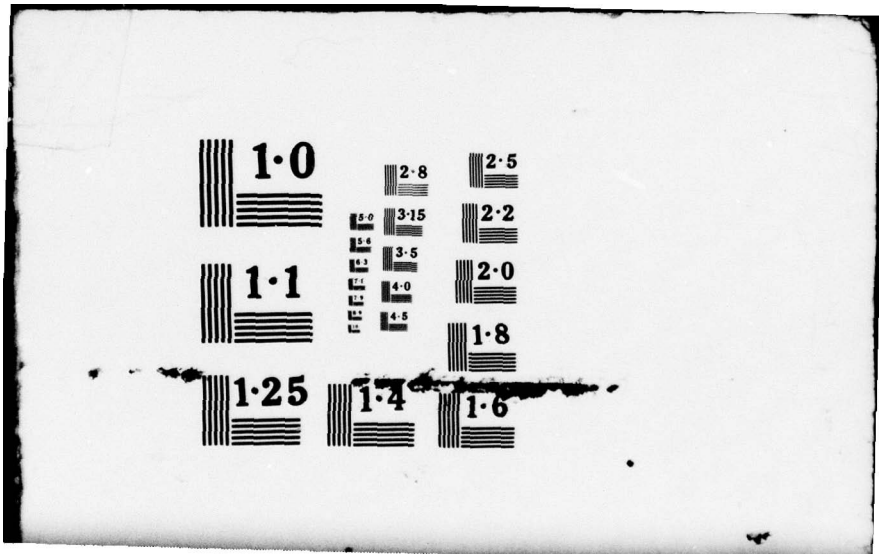
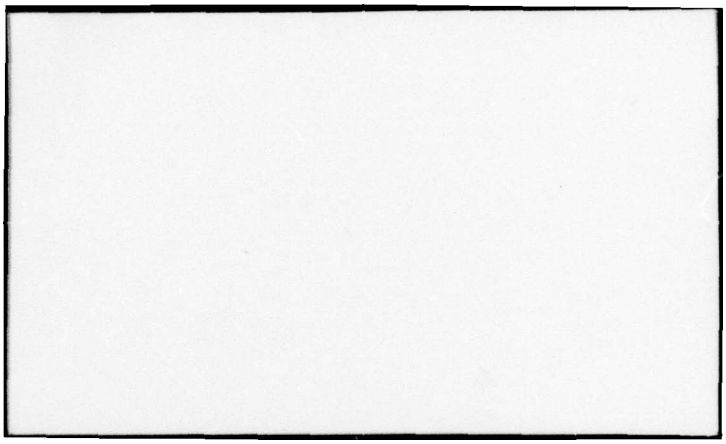
AD-A067 247 WHARTON SCHOOL PHILADELPHIA PA DEPT OF DECISION SCIENCES F/G 5/1
DBALERT: AN ALERTING SYSTEM FOR WAND.(U)
DEC 78 J S RIBEIRO N00014-75-C-0440
UNCLASSIFIED 78-12-02 NL

END
DATE FILMED
6-79
DDC


1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8
3·15
3·5
4·0
4·5



(12) (9)

DBALERT: AN ALERTING SYSTEM FOR WAND

James S. Ribeiro

78-12-02

Department of Decision Sciences
The Wharton School
University of Pennsylvania
Philadelphia, PA 19104

Research supported in part by the Office of Naval Research under Contract N00014-75-C-0440.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: 78-12-02	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): DBALERT: An Alerting System for WAND.	5. TYPE OF REPORT & PERIOD COVERED: Technical Report. Jan 1978 — Dec 1978,
6. PERFORMING ORG. REPORT NUMBER: 78-12-02	
7. AUTHOR(s): James S. Ribeiro	8. CONTRACT OR GRANT NUMBER(s): N0001 4-75-C-0440
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Department of Decision Sciences, University of Pennsylvania, Philadelphia, PA 19104	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: Task NR049-360
11. CONTROLLING OFFICE NAME AND ADDRESS: Office of Naval Research	12. REPORT DATE: December 1978
	13. NUMBER OF PAGES: 15
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper describes DBALERT, which performs simple alerting for the WAND DBMS. A demonstration is shown using DBALERT to perform alerting on the ONRODA and SYSFIL databases.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601 |

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ABSTRACT

Tnis paper describes DBALERT, which performs simple alerting for the WAND DBMS. A demonstration is shown using DBALERT to perform alerting on the ONRODA and SYSFIL databases.

1.0 Introduction

DBALERT extends DBLOOK by providing interactive alerting capabilities. Alerting is the ability to monitor databases for conditions of interest to the user. This is accomplished in DBALERT by allowing the user to interactively define alerters, which will monitor the updates to the database. DBALERT also allows the user to specify information which will be reported back if the condition ever becomes true.

1.1 Alerting Capabilities

DBALERT allows simple alerters to be defined for additions, deletions, or modifications to the database. Simple alerters monitor conditions which can be defined in terms of a single record class. An example of a simple alerter is the request

"Report any savings account number if the balance of the account drops below $400.00."

where savings account number and savings balance are within the same record type.

Alerters monitor updates to the database rather than its present state. Conditions are never evaluated when an alerter is just defined and are only evaluated as true when an update occurs and a condition becomes true which was not

previously true. Therefore, if the balance dropped from $450 to $380 to $360 then only the drop to $380 would trigger an alerter.

It is conceivable that the user may want to be alerted whenever there is an update and the condition is true. In this case field names within the condition can be prefixed with the timed variables OLD: or NEW:. Therefore, in the previous example if the balance drops from $450 to $380 to $360 then both drops to $380 and $360 will trigger alerters.

1.2 Using DBALERT

In order to use DBALERT the user must have specified the ALERTED DATABASE clause in the Schema Entry. If this clause was not specified when the schema was created then a new schema and sub-schema must be created with this clause included in the Schema Entry. The database does not have to be reloaded. Inclusion of this clause will increase the size of the schema and the running times of programs. Therefore, the ALERTED DATABASE clause should only be included when the user is interested in monitoring the database. To run DBALERT type at monitor level

```
X DBALERT
```

At this point any of the HELP commands can be entered.

1.3 HELP in DBALERT

HELP in DBALERT extends the help facility of DBLOOK by providing help for all DBALERT commands.

1.4 DBALERT Commands

DBALERT contains the following types of commands:

1. The ADDALERT command to define new alerters.
2. Commands to manipulate existing alerters and obtain information about them.

1.5 ADDALERT

The ADDALERT command is of the form:

ADDALERT alerter-name type condition report-list

'Alerter-name' is a user-defined name which is used to identify the alerter. 'Alerter-name' must be unique and is from 1 to 30 characters in length.

'Type' specifies the alerter type and may be either IFADD, IFDEL, or IFMOD. IFADD implies that this alerter should be triggered if a record is added to the database

which satisfies the specified condition. IFDEL will trigger an alerter if a record with the condition is deleted from the database and IFMOD triggers the alerter if any data which is modified meets the condition.

'Condition' specifies the condition which DBALERT is to monitor. A condition can contain literals, field names, arithmetic operators, relational operators, and logical operators.

Field names may be prefixed with optionally specified time qualification (OLD: or NEW:), where OLD: and NEW: refer to the pre and postupdate values of an item in a record whose update triggers an alert.

A literal can be an integer, a real number, a double precision number, or a character string. Character strings must be enclosed within quotes (').

The following arithmetic operators are acceptable:

**	exponentiation
*	multiplication
/	division
+	addition (infix)
	or designation of sign (unary)
-	subtraction (infix)
	or negation (unary)

These operators are used to combine numeric data-items and numeric literals into arithmetic expressions. Exponentiation has the highest precedence, followed by multiplication and division (which have equal precedence), then unary plus and negation (which have equal precedence), and finally addition and subtraction (which also have equal precedence).

The relational operators consist of:

EQ (=)	equal to
NE	not equal to
GT (>)	greater than
GE	greater than or equal to
LT (<)	less than
LE	less than or equal to

Relational operators combine data-items, literals, or arithmetic expressions into relational expressions. Relational operators are of equal precedence and are lower in precedence than all arithmetic operators.

DBALERT accepts the logical operators AND, OR, and NOT. Logical operators are used to combine relational expressions and logical expressions into logical expressions. Logical operators are lower in precedence than both arithmetic operators and relational operators. The NOT operator has precedence over AND and then OR.

A condition is evaluated in the order of precedence of the operators. If there is more than one operator at the same precedence level then they are evaluated from left to right. Parentheses can be used to change the order of evaluation.

A condition can also be the name of a record. In this case, DBALERT will monitor the database for additions, deletions, or modifications to the specified record.

'Report-list' contains a list of field names or character strings which are reported back if the alerter is triggered. The report-list can contain the keywords DATE and TIME which report the date and time the alerter was triggered.

1.6 LIST

The command LIST will list the names of all the alerters.

1.7 ERASE

Alerters can be deleted by using the ERASE command.

ERASE alerter-name

A similar command is ERASE ALL which deletes all the defined alerters.

1.8 DEACTIVATE

An alerter is made inactive by the DEACTIVATE command. An inactive alerter can never be triggered.

DEACTIVATE alerter-name

DEACTIVATE ALL deactivates all defined alerters.

1.9 ACTIVATE

The ACTIVATE command is used to activate an inactive alerter.

ACTIVATE alerter-name

All alerters can be activated by the command ACTIVATE ALL. An alerter is active on being created.

1.10 STATE

The STATE command will list the current state of an alerter.

STATE alerter-name

The following information is provided by the STATE command:

1. The number of times this alerter has been triggered.
2. The date this alerter was last triggered.
3. The time this alerter was last triggered.
4. Whether this alerter is inactive or active.

STATE ALL will display this information for all defined alerters.

1.11 Demonstration

This section contains an actual terminal session with DBALERT. In this demonstration, both the SYSFIL and ONRODA databases are monitored.

```
.X DBALERT

Welcome to B.07 DBLOOK and SEED.  Type HELP or ? for info.
*       WATCH
Which database would you like to watch? SYSFIL
```

```
*       *LOOK AT SYSREC RECORD IN SYSFIL DATABASE

*       SYSREC

RECORD NAME IS SYSREC
    WITHIN AREA1
    LOCATION MODE IS CALC USING JOBNUMBER
      DUPLICATES ARE NOT ALLOWED.
CONTAINS THE FOLLOWING ITEMS:
JOBNUMBER
USERNAME
TTY
PROGNAME
PROJECTNUMBER
PROGRAMMERNUMBER
*       *NOW DEFINE SOME ALERTERS

*       ADDALERT

Alerter name

*       MONITOR_DBALERT_USERS

Alerter type

*       IFADD

Condition

*       PROGNAME EQ 'DBALER'

Report list

*       USERNAME ' IS USING DBALERT'

*       ADDALERT MONITOR_BUNEMAN IFMOD

Condition

*       USERNAME = 'BUNEMA' AND PROGNAME EQ 'PASCAL'

Report list

*       'PETER BUNEMAN IS USING PASCAL'

*       LIST

MONITOR_DBALERT_USERS

MONITOR_BUNEMAN

*       *NOW LETS UPDATE THE SYSFIL DATABASE

*       UPDATE
```

```
Enter frequency of checking (integer seconds) : 30

Total monitoring time ? (integer minutes): 1

ALERT MONITOR_DBALERT_USERS
RIBEIJ  IS USING DBALERT
*       WATCH

Which database would you like to watch? ONRODA

*       *DELETE ALL DEFINED ALERTERS

*       ERASE ALL

Confirm(Y OR N):Y

All alerters deleted
*       *LETS DEFINE SOME ALERTERS FOR THE ONRODA DATABASE

*       ADDALERT BESSMENY_SIGHTED IFADD

Condition

*       TRACK-NAME EQ 'BESSMENY'

Report list

*       'THE BESSMENY WAS SIGHTED'

*       ADDALERT RED-CONTACT-FLAG IFADD

Condition

*       CONTACT-FLAG = 'RD'

Report list

*       CONTACT-NAME 'WAS SIGHTED WITH RED FLAG'

*       ADDALERT

Alerter name

*       LOW_FUEL_RESERVE

Alerter type

*       IFADD

Condition

*       SHIP-PERCENT-FUELED < 65

Report list
```

```
*       'FUEL RESERVES HAVE DROPPED BELOW 65% FOR' STATUS-SHIP-NAME

*       U

Enter name of file containing database updates: UPD1.DAT

Enter time compression factor: 8000

Track-record being stored @ 7702111830
         Track-number =     4
         Track-name = BESSMENY
         Track-force =      7
ALERT BESSMENY_SIGHTED
THE BESSMENY WAS SIGHTED

Contact-report record now being stored @ 7702111830
         Contact-name = BESSMENY
         Contact-flag = RD
         Contact-course = *****
         Contact-speed = *****
         Contact-altitude =
         Contact-latitude = 31-00N
         Contact-longitude = 26-00E
         Contact-dtg = 7702111830
         Contact-semi-major-axis =    10
         Contact-semi-minor-axis =    10
         Contact-angle = *****
         Contact-emitter =
ALERT RED-CONTACT-FLAG
BESSMENY    WAS SIGHTED WITH RED FLAG

Track-record being stored @ 7702111830
         Track-number =    10
         Track-name = DODRY
         Track-force =      7

Contact-report record now being stored @ 7702111830
         Contact-name = DODRY
         Contact-flag = RD
         Contact-course = *****
         Contact-speed = *****
         Contact-altitude =
         Contact-latitude = 30-40N
         Contact-longitude = 26-40E
         Contact-dtg = 7702111830
         Contact-semi-major-axis =    10
         Contact-semi-minor-axis =    10
         Contact-angle = *****
         Contact-emitter =
ALERT RED-CONTACT-FLAG
DODRY  WAS SIGHTED WITH RED FLAG

Defence-condition-record now being modified @ 7702151645
         Defence-condition-id =     1
         Defence-condition =     3
```

```
Weather record now being stored @ 7702151730
         Weather-dtg = 7702151730
         Weather-precip-type = RN
         Weather-precip-rate = 1.100

Equipment-casualty-report now being stored @ 7702151745
         Equipment-id = CG-11
         Equipment-casualty-report-dtg = 7702151745
         Equipment-name = RIM-24
         Equipment-readiness = 1

Ship-status-report record now being stored @ 7702151930
         Ship-status-report-dtg = 7702151930
         Ship-percent-fueled =    60
         Status-ship-name = COCHRANE
ALERT LOW_FUEL_RESERVE
FUEL RESERVES HAVE DROPPED BELOW 65% FOR  COCHRANE

Track-record being stored @ 7702152020
         Track-number =     2
         Track-name = UNK
         Track-force =     5

Equipment-casualty-report now being stored @ 7702152030
         Equipment-id = DD-964
         Equipment-casualty-report-dtg = 7702152030
         Equipment-name = SONAR
         Equipment-readiness = 3
*       EXIT

STOP

END OF EXECUTION
CPU TIME: 22.57 ELAPSED TIME: 3:19.08
EXIT
```

REFERENCES

1. Buneman, O.P., and H.L. Morgan, "Alerting Techniques in Database Systems", Proceedings of IEEE Compsac Conference, Chicago, 1977.

2. CTEC Inc., "An ONRODA Scenario for Demonstrating Alerts", Falls Church, Va.

3. Gerritsen, R., J. Ribeiro, R. Cortes, and R. Zowader, WAND User's Guide, Working Paper No. 76-01-03, Department of Decision Sciences, The Wharton School.

4. Ribeiro, J., A Database Interface to WAND for the Network Alerter Service, M.S. Thesis, Moore School, U. of Pa.